

# How does that Work?

## Error-correcting codes

Many types of error-correcting techniques exist, but in data communications, Hamming encoding probably finds the widest use. Unlike serial-port communications that use a single parity bit, Hamming encoding adds many additional check bits to a transmission. Those bits take extra time to transmit, but the added overhead brings about more reliable communications. As used today, basic Hamming encoding will let a receiver detect and correct a single-bit error and detect—but not correct—a two-bit error. Luckily, transmission errors in packets of a few bytes usually involve only a single bit. (The theory behind Hamming codes goes beyond this discussion.)

To find the number of Hamming check bits,  $k$ , you need to correct for a single error in a group of  $m$  bits, using the formula:

$$m \leq 2^k - k - 1$$

Thus, for a 16-bit word, five bits provide single-error correction. By adding another bit, for a total of six, you'll obtain single-error correction and double-error detection.

The example in the **Figure** shows a 16-bit word and the bits in it used to compute six check bits. Simply sum the marked bits in each row to get the respective check bit. For example:

$$CB_0 = D_{13} + D_{10} + D_9 + D_8 + D_4 + D_3 + D_1 + D_0$$

(An odd number of 1's sums to 1, an even number of 1's

sums to 0.) Thus, calculating the checkbits as shown in Figure 1, the data word ( $D_{15}$ – $D_0$ ) 0011 1101 1001 1001 yields check bits ( $CB_5$ – $CB_0$ ) 101100, which get transmitted with the 16 data bits.

How does the receiver use the check bits to locate an error? Assume the reception of data 0100 1101 0100 0111 and check bits 010110. The receiver calculates a new set of check bits—000000—based on the received data. In this example, the received check bits and the newly calculated check bits show differences at  $CB_4$ ,  $CB_2$ , and  $CB_1$ . The figure shows that only data bit  $D_5$  influences these three check bits, so that bit must be incorrect, or inverted. Thus, the actual data should be 1011 1101 0110 0111. Each bit in the data word influences only three of the check bits, so a single-bit error will produce *three* incorrect check bits.

In a communication system, hardware would compute new check bits and compare them with the received check bits. The comparator's output would address a simple look-up table that would identify the incorrect bit so the receiver could invert it.

The check bits also yield other useful information. A single check-bit difference indicates an error in the check bits themselves. An even number of check-bit differences indicates two bit errors, which a receiver cannot correct. Three or more bit errors can fool some error-detecting and error-correcting systems. Thus, some errors will require a retransmission of data.—Jon Titus

Check bit	16-bit data															
	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
CB0			•			•	•	•				•	•		•	•
CB1		•			•			•		•	•		•	•		•
CB2	•			•			•		•		•	•		•	•	
CB3				•	•	•			•	•				•	•	•
CB4	•	•	•						•	•	•	•	•			
CB5	•	•	•	•	•	•	•	•								

"How does that work?" is a bimonthly column covering test or measurement products that take advantage of a novel effect or unusual technology.